Redmine - Feature #5390

customize terminology

2010-04-27 19:51 - Daniel Miller

Status:	Closed	**Start date:**	2010-04-27
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Plugin Request	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Wont fix		

Description

Some schools of thought call the stuff that Redmine stores by different names: defects, bugs, issues, (trouble-)tickets. The basic terminology used by Redmine should be stored in an SQL table that is queried at the time of starting up Redmine. Redmine would then use these strings instead of the current "issue". There would be a page in administrator mode to change these strings in this terminology table. Various stings used in the Redmine UI may be considered customizable terminology, whereas other strings may be noncustomizable terminology. A customized term would be used not only in the browser-based UI, but also in emails and all other mentionings of that term.

Some people might consider this which-string-to-display problem to be related to localization and internationalization, but here the US/en remains the same, so it is not exactly the same thing.

History

#1 - 2010-04-27 20:21 - Jan Schenck

+1

#2 - 2010-04-27 20:50 - Casper Valdemar Poulsen

I recently had to change terminology (issues to tasks, versions to milestones etc.) to better fit into the general project management style at our company, and what I did was just to patch the english localization files, en.yml. Pretty much just search and replace and works really well. The only place to tell what's changed, is in the URL.

#3 - 2010-04-27 23:00 - Felix Schäfer

Whatever you call them, storing recurrent strings-to-be-displayed in the DB is a bad idea. I think the best way to go for this would be to make plugins: you can override localization strings in them. Make a plugin to suit your terminology, that way you can share your changes and be somewhat upgrade-stable.

#4 - 2010-05-12 05:38 - Eric Davis

- Category set to Plugin Request

- Status changed from New to Closed

- Resolution set to Wont fix

Felix Schäfer wrote:

> Whatever you call them, storing recurrent strings-to-be-displayed in the DB is a bad idea.

I agree, hitting the database for every text on the page would add a significant database load to each page request.

I've created a plugin back in January that shows how you can customize Redmine's locale strings. Just install a copy of it and follow the Readme. http://github.com/edavis10/redmine_language_change (if you install it to vendor/plugins/zzz_redmine_language_change, it will be loaded last and you can override any plugin stings.)

If you **really** must have the locale strings in the database, we are just using the Rails i18n system. You can probably swap out the storage backends with a bit of work, see http://rails-i18n.org to get started.

#5 - 2010-05-12 07:38 - Daniel Miller

Eric Davis wrote:

> Felix Schäfer wrote:

> > Whatever you call them, storing recurrent strings-to-be-displayed in the DB is a bad idea.
>
> I agree, hitting the database for every text on the page would add a significant database load to each page request.

I did not say "hit the database for *every ... page*". I said "SQL table that is queried at the *time of* ***starting up*** *Redmine*". Those are quite different concepts, unless you are in fact (inefficiently) "starting up Remine" afresh "for every page".

> I've created a plugin back in January that shows how you can customize Redmine's locale strings. Just install a copy of it and follow the Readme. http://github.com/edavis10/redmine_language_change (if you install it to vendor/plugins/zzz_redmine_language_change, it will be loaded last and you can override any plugin stings.)

This suggestion is useful. Thanks. But quite honestly a particular industry's choice of words should not be a plugin any more than the Estonian language should be. The localization tuple should be (nation, language, culture) not merely (nation, language), not merely in Redmine, but in all software. In fact, often the (language, culture) partial tuple (e.g., same industry throughout the English-speaking nations) is a stronger, unifying single localization than (nation, language).